IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: GYUGYI, et al.	§	Confirmation No.: 4738
Serial No.: 10/731,602	§	Group Art Unit: 2609
Filed: December 9, 2003	§	Examiner: Maglo, Emmanuel K.
For: STORING AND ACCESSING TCP CONNECTION INFORMATION	§	

MAIL STOP AMENDMENT
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

SUPPLEMENTAL AMENDMENT AND INTERVIEW SUMMARY

Dear Sir:

This amendment is submitted following an interview with the Examiner on October 29, 2008 and the filing of a Response to the Office Action dated June 4, 2008. Please enter this response and reconsider the claims pending in the application for reasons discussed below. Although Applicant believes that no additional fees are due in connection with this response, the Commissioner is hereby authorized to charge counsel's Deposit Account No. 20-0782/NVDA/P000860/SW for any fees, including extension of time fees or excess claim fees, required to make this response timely and acceptable to the Office.

Amendments to the Claims are reflected in the listing of claims which begins on page 2 of this paper. **Remarks** begin on page 9 of this paper.

IN THE CLAIMS:

The following listing of claims will replace all prior listings of claims in the application:

1. (Currently Amended): A method of using a delegated connection table, comprising:

selecting, by a transmission control protocol (TCP) stack, a connection for processing by an offload unit;

initializing an entry in the delegated connection table with connection state corresponding to the connection selected by the transmission control protocol (TCP) stack for processing by the offload unit;

determining that a first frame is received on the connection selected by the TCP stack for processing by the offload unit;

updating the entry when [[a]] the first frame is received for the connection, wherein a sequence number ~~in the first frame~~ is stored in the entry, the sequence number representing a next expected sequence number for the connection;

parsing the first frame to extract TCP payload data;

uploading the TCP payload data to a memory; and

reading the entry when a second frame is transmitted for the connection.

2. (Previously Presented): The method of claim 1, further comprising updating the entry by copying a portion of the second frame into a portion of the entry in the delegated connection table when the second frame is transmitted.

3. (Cancelled)

4. (Previously Presented): The method of claim 1, further comprising uploading payload data to a location specified in the entry within a memory space of the memory that is allocated to an application program.

5. (Previously Presented): The method of claim 1, further comprising notifying the TCP stack when the TCP payload data of the first frame received is uploaded by the offload unit to at least one legacy buffer that is in a portion of the memory that is allocated to a driver configured to interface between the offload unit and an application program.

6. (Previously Presented): The method of claim 1, wherein the TCP payload data of the first frame is uploaded to a legacy buffer that is in a portion of the memory that is allocated to a driver configured to interface between the offload unit and an application program.

7. (Previously Presented): The method of claim 1, further comprising:

receiving a third frame that does not correspond to another entry in the delegated connection table; and

uploading the third frame to a legacy buffer that is in a portion of the memory that is allocated to a driver configured to interface between the offload unit and an application program.

8. (Previously Presented): The method of claim 1, further comprising:

determining that the first frame and the second frame are out-of-sequence based on a comparison of the sequence number stored in the entry with a sequence number in the second frame; and

storing a flag in the entry to indicate that synchronization is requested for the connection.

9. (Cancelled)

10. (Previously Presented): The method of claim 1, further comprising uploading the payload data of the first frame to at least one legacy buffer that is in a first portion of the memory that is allocated to a driver configured to interface between the offload unit and an application program when a user buffer in a second portion of the memory that is allocated to the application program is not available.

11. (Previously Presented): The method of claim 8, further comprising notifying the application program to complete processing of the second frame.

12. (Previously Presented): The method of claim 8, further comprising uploading any subsequent frames received for the connection, to one or more additional legacy buffers, until resynchronization is signaled by the TCP stack.

13. (Previously Presented): The method of claim 12, wherein the resynchronization is accomplished by sending an acknowledge for the second frame, and invalidating any buffer descriptors for portions of the memory that are available for storing data received on the connection.

14. (Previously Presented): The method of claim 12, further comprising:

determining that the sequence number in the second frame is more advanced than the sequence number stored in the entry;

sending an acknowledge for the first frame; and

invalidating any buffer descriptors for portions of the memory that are available for storing data received on the connection.

15. (Previously Presented): A method of accessing a delegated connection table during processing of a received frame, comprising:

reading a connection match portion of the delegated connection table, wherein the connection match portion of the delegated connection table stores delegated connections that are selected, by a transmission control protocol (TCP) stack, for processing by an offload unit that includes the delegated connection table;

determining the received frame corresponds to an entry in the connection match portion of the delegated connection table;

reading a connection data portion of the delegated connection table that stores an expected sequence number, an acknowledgment (ACK) number, timestamp data, and a count of unACKnowledged frames in the entry; and

parsing the received frame to produce payload data.

16. (Original): The method of claim 15, further comprising:

modifying a portion of connection state data stored in the connection data portion of the delegated connection table.

17. (Previously Presented): The method of claim 15, further comprising:

reading a connection buffer portion of the delegated connection table to obtain user buffer information including a user buffer address and a corresponding user buffer length of a user buffer that is stored in a portion of memory allocated to an application program.

18. (Previously Presented): The method of claim 17, further comprising:

determining the user buffer information indicates a user buffer is not available; and

requesting a user buffer by setting a request buffer flag in the connection buffer portion of the delegated connection table.

19. (Original): The method of claim 17, further comprising uploading the payload data to the user buffer.

20. (Previously Presented): The method of claim 18, further comprising:

determining a receive buffer has reached a high water mark; and

uploading the payload data to a legacy buffer that is in a portion of the memory that is allocated to a driver configured to interface between the application program and the offload unit including the delegated connection table.

21. (Previously Presented): The method of claim 18, further comprising:

determining a buffer request timer has expired; and

uploading the payload data to a legacy buffer that is in a portion of the memory that is allocated to a driver configured to interface between the application program and the offload unit including the delegated connection table.

22. (Previously Presented): A delegated connection table for storing delegated connection information, comprising:

a first storage resource configured to store user buffer information for delegated connections including a user buffer length and a user buffer address corresponding to a portion of memory that is allocated to an application program;

a second storage resource configured to store delegated connection state information for the delegated connections including an expected sequence number, an acknowledgment (ACK) number, timestamp data, and a count of unACKnowledged frames; and

a third storage resource configured to store delegated connection identification information for the delegated connections including a destination IP address, a source IP address, a source transmission control protocol (TCP) port, and a destination TCP port, wherein the delegated connections are selected, by a transmission control protocol (TCP) stack, for processing by an offload unit that includes the delegated connection table.

23. (Cancelled)

24. (Original): The delegated connection table of claim 22, further comprising a command processing unit configured to write to the first storage resource.

25. (Previously Presented): The delegated connection table of claim 22, further comprising a transmit engine configured to access the second storage resource and perform outbound frame processing.

26. (Previously Presented): The delegated connection table of claim 22, further comprising a receive engine configured to access the second storage resource and parse incoming frames and determine whether or not the incoming frames are valid.

27. (Original): The delegated connection table of claim 26, wherein the receive engine is configured to read the first storage resource.

28. (Original): The delegated connection table of claim 26, wherein the receive engine is configured to read the third storage resource.

29. (Previously Presented): The method of claim 1, wherein the updating of the entry when the first frame is received for the connection includes clearing an unACKnowledged count, updating an acknowledgment (ACK) number with a last ACKnowledged number, and updating the sequence number with an incremental sequence number that is stored in the entry.

30. (Previously Presented): The method of claim 16, wherein the modifying of the portion of the connection state data includes clearing an unACKnowledged count, updating the acknowledgment (ACK) number with a last ACKnowledged number, and updating the expected sequence number with an incremental sequence number.

31. (Previously Presented): The delegated connection table of claim 22, wherein the delegated connections specified by the delegated connection table are a subset of active connections stored in a connection table within a system memory.

REMARKS

This amendment is submitted pursuant to the interview of October 29, 2008 between the Examiner Emmanuel Maglo and Agent of Record Stephanie Winner. Stephanie Winner explained the invention and discussed the amendments made to claims 1, 15, and 22 with Examiner Maglo. In particular, the limitations of selecting a connection for processing by the offload unit and storing the sequence number in the entry were discussed. Examiner Maglo advised Stephanie Winner to review column 15, lines 14-20 regarding the limitation of storing a sequence number. Applicants respectfully request reconsideration and allowance of all claims in view of the following remarks.

Amendments

As amended, claim 1 recites the limitations of determining that a first frame is received on the connection selected by the TCP stack for processing by the offload unit and storing a next expected sequence number for the connection in an entry of the delegated connection table. As supported in Figure 4B and paragraphs [0055] – [0056] of the application as originally filed, the TCP stack determines whether or not a frame was received on a delegated connection that is selected by the TCP stack for processing by the offload unit. The TCP stack may select a connection for processing by the offload unit for a variety of reasons as described in paragraph [0053] of the application as originally filed. As supported in paragraph [0049] of the application as originally filed, a next expected sequence number is stored as a portion of the acknowledgement state in the entry of the delegated connection table.

The Elzur reference fails to teach each and every limitation recited in amended claim 1. The TCP offload apparatus of Elzur processes all TCP packets, buffers portions of the packets in elastic buffers, and places portions of the packets in host memory. Nowhere does Elzur teach or suggest that a connection is selected by a TCP stack for offload processing, as claimed. In contrast, as the reference makes clear in Figure 11 step 120, each and every incoming TCP packet is associated with a connection. In particular, Elzur fails to teach or suggest the limitation of determining that a frame is received on the connection selected by the TCP stack for processing by the offload unit.

Elzur also fails to teach or suggest the limitation of storing a sequence number representing a next expected sequence number in an entry of the delegated connection table, as recited in amended claim 1 and claims 15, and 22. In Figure 11 step 130 TCP connection context is fetched for a received frame. Elzur teaches that context information is stored in a memory and accessed to process frames for a connection (see col. 10, lines 46-48 and lines 59-65 and Figures 12, 13, and 14). However, Elzur does not describe the specific connection state that is included in the context information. In column 15, lines 14-20 Elzur describes constructing a mapping between TCP sequence numbers and the host buffers. As shown in Figure 9, the mapping occurs through a buffer descriptor table and is used as an offset to a base sequence number to determine a physical address of a host buffer. Nowhere does Elzur teach or suggest storing a next expected sequence number in an entry of a delegated connection table.

Since Elzur fails to teach each and every limitation recited in amended claim 1, this claim and dependent claims 2, 4-8, 10-14, and 29 cannot be anticipated by Elzur.

CONCLUSION

Based on the above remarks, Applicants believe that they have overcome all of the rejections set forth in the Office Action dated June 4, 2008, and that the pending claims are in condition for allowance. If the Examiner has any questions, please contact the Applicant's undersigned representative at the number provided below.

Respectfully submitted,

Stephanie Winner
Registration No. 52,371
PATTERSON & SHERIDAN, L.L.P.
3040 Post Oak Blvd. Suite 1500
Houston, TX 77056
Telephone: (713) 623-4844
Facsimile: (713) 623-4846
Agent for Applicants